Attorney Docket No. 08049.0001

United States Patent Application

of

Leo J. Campbell

Jon L. Cook

Charles R. Chamberlain

Michael J. McGrath

and

Isadore Schoen

for

Systems and Methods for Authenticating an Electronic Message

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of provisional application "Apparatus and Method for Authenticating Digital Messages and Other Files," filed September 30, 1999 and assigned Serial No. 60/157,168, and provisional application "Systems and Methods for Establishing an Electronic Account and Providing Services in a Network," filed March 17, 2000 and assigned Serial No. 60/189,983. The contents of the above applications are relied upon and expressly incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to systems and methods for providing electronic communication services to customers. More particularly, the invention relates to systems and methods for providing content and temporal integrity and identification verification to electronic messages shared by users over a network.

Description of the Related Art

The use of electronic networks to convey information among networked users has undergone an enormous amount of growth in recent years. The ability to transfer data using computer applications, such as, for example, e-mail and file transfer protocol programs, has become increasingly important in personal, and especially, business communications.

Using computer networks for business communications, including buying and selling goods online, electronic funds transfer, online advertising, and accessing business information resources is known as electronic commerce (E-commerce). E-commerce can improve the efficiencies of current business processes and provide opportunities to widen existing customer bases. Over the next few years, as the number of Internet users continues to expand, E-commerce has the potential to be the source of an extraordinary amount of revenue growth.

In order to realize this potential, a variety of communication services and features will be required for E-commerce which traditionally have been available in physical communication channels. The United States Postal Service (USPS), an independent establishment of the executive branch of the U.S. government, provides such features through a variety of document and package delivery services. The USPS is widely recognized as a secure and reliable means for sending and receiving packages and mail used for both personal and business transactions. Packages and mail sent via the USPS are time-stamped with an official postmark which provides the recipient proof of the time the item was sent. Additionally, once material is placed with the USPS, the document is no longer in the sender's control, and thus cannot be recalled. Furthermore, packages and mail sent through the USPS are protected from third-party tampering by Federal

LAW OFFICES
FINNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

laws. Electronic communication services currently do not provide these features. Additional security enhancements, such as authenticating the identities of the parties involved in a transaction and/or providing assurance to the recipient that a received message has not been altered may also be required for E-commerce to reach its full potential.

To ensure the vitality and growth of electronic communication and commerce, consumers and businesses need a secure way to communicate and conduct business electronically. Without trustworthy channels of communication, many potential participants in electronic commerce may be unwilling to send sensitive information electronically. In light of the foregoing, it is desirable to provide a system for electronic communication that provides a level of security which meets or exceeds the current level offered by the existing physical mail and package delivery services.

SUMMARY OF THE INVENTION

In accordance with the purpose of the present invention, as embodied and broadly described herein, the invention provides methods and apparatuses for authenticating an electronic message. The electronic message containing data and an electronic address is received from a sender. A digest is then created based on the message data and a temporal stamp is appended to this digest. The digest and temporal stamp are signed using a digital signature. The digest,

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

the temporal stamp, and the digital signature are then sent to an electronic address and thereafter authenticated.

Exemplary systems and methods consistent with the present invention are recited in the attached claims. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate several embodiments of the invention and together with the following description, serve to explain the principles of the invention. In the drawings:

Fig. 1 is a flow chart of a method consistent with the present invention.

Fig. 1A is a simplified block diagram of first operational mode for a system to authenticate electronic messages in accordance with the present invention;

Fig. 1B is a simplified block diagram of a second operational mode for a system to authenticate electronic messages in accordance with the present invention;

Fig. 2A is a simplified block diagram of the first embodiment for a system to authenticate electronic messages in accordance with the present invention;

LAW OFFICES
FINNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

Fig. 2B is a simplified block diagram of the second embodiment for a system to authenticate electronic messages in accordance with the present invention;

Fig. 3A is block diagram showing components of a third embodiment for a system in accordance with the present invention;

Fig. 3B is a detailed block diagram showing components of a fourth embodiment for a system in accordance with the present invention;

Fig. 3C is a detailed block diagram showing a fifth embodiment for a system in accordance with the present invention;

Fig. 4A is a data-flow diagram corresponding to the embodiment of Fig. 3A;

Fig. 4B is a data-flow diagram corresponding to the embodiment of Fig. 3B;

Fig. 4C is a data-flow diagram corresponding to the embodiment of Fig. 4C;

Fig. 5A is a block diagram showing software modules responsible for inter-client interaction of the embodiment of Fig 3A;

Fig. 5B is a block diagram showing software modules responsible for inter-client interaction of the embodiment of Fig 3B;

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

Fig. 6 is a simplified block diagram showing hardware and software components of the server of Fig. 3A;

Fig. 7 is a detailed block diagram showing the hardware components corresponding to the embodiment of a Fig. 3A;

Fig 8A is a data processing diagram corresponding to one processing mode in accordance with the present invention; and

Fig. 8B is data processing diagram corresponding to another processing mode in accordance with the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference will now be made in detail to the present preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Whenever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Fig. 1 shows a flow chart of a method consistent with the present invention. A sender will generate an electronic message, containing message data and a destination electronic address, for which authentication is desired. The sender submits the electronic message over a network to an EPM (Electronic PostMark) system for authentication. The EPM system then receives the electronic message from the sender (Step 50). The EPM system then

LAW OFFICES
FINNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

creates a tag or "digest" from the electronic message (Step 52). The digest is a digitally compressed representation unique to the electronic message. The EPM system then appends a temporal stamp to the digest (Step 54). The temporal stamp includes the time and date denoting when the temporal stamp was applied. The digest and the temporal stamp are then signed by the EPM server using a digital signature (Step 56). The digital signature, digest, and temporal stamp are sent by the EPM server over the network to the destination electronic address (Step 58). The digital signature, digest, and temporal stamp are then authenticated (Step 60). The authentication process typically takes place on the data processing machine at the electronic address; however, the authentication could take place on a different data processing machine.

Fig. 1A shows a simplified block diagram of a first mode of operation for a system consistent with the present invention. A sender 110 generates the electronic message, including the destination electronic address, for transfer over a network 115 which may be a public network such as the Internet. The message can be a digital document in any type of format. The electronic message, along with the destination electronic address, is received by an EPM system 120. EPM system 120 then generates an EPM data structure. The EPM data structure includes the digest and the temporal stamp. The digest and temporal stamp are digitally signed by the EPM system 120, and the resulting

LAW OFFICES
FINNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

digital signature is also included in the EPM data structure. As known to those skilled in the art, a digital signature is extra data appended to input data which authenticates both the input data and the identity of any signer over the input data.

For this invention, the digital signature ensures the authenticity of the EPM data structure and the identity of the EPM system 120. Any unauthorized modifications to the temporal-stamp or the digest can be detected through examining the digital signature. Furthermore, any alterations in the electronic message itself may be detected though examining the digest. Therefore, the EPM data structure can afford at least three assurances for an electronic transaction. The first is the electronic message existed at a known point in time. The second assurance is the identification of EPM system 120 is known to a recipient 130 of the electronic message. The third assurance is alteration of the contents of the electronic message as received by recipient 130 is detectable after the generation of the EPM data structure.

Further referring to Fig. 1A, the recipient 130 typically receives the EPM data structure and the electronic message over network 115 from EPM server 120. Recipient 130 may then verify the identity of the signer of the EPM data structure and the integrity of the electronic message using the EPM data structure and verification application. The verification application is discussed in more detail below.

Sender 110 may elect to have EPM system 120 forward only the EPM data structure to recipient 130 and may then send the electronic message itself to recipient 130 directly over the network. This procedure provides sender 110 with more control in how the electronic message is routed through EPM system 120, as will be described below.

Each time an EPM data structure is generated, it is stored in a log located within EPM system 120. This log provides an actual record which can be used to prove that an EPM data structure was generated for a given message. One advantage of the invention is that electronic messages which have been postmarked by EPM system 120 may be afforded legal protections under laws which protect official entities, such as the USPS. Therefore, EPM data structures stored in the log file can serve as legal proof of the existence and digest of an electronic message. An EPM data structure of the log file itself may be generated to insure the integrity of the log file.

Fig. 1B shows a simplified block diagram of a second mode of operation for a system consistent with the present invention. In this mode of operation, sender 110 and recipient 130 are the same entity. Sender 110 prepares an electronic message and submits it via network 115 to EPM system 120 in the same manner as described above. However, in this embodiment, the electronic address is included with the electronic message is that of sender 110. EPM system 120 generates an EPM data structure as before, and then typically returns only the EPM data structure back to sender 110. This mode of operation allows sender 110 to use EPM system 120 as a type of electronic verification service, whereby sender 110 can validate the existence of an electronic message at a specific point in time and the contents of the message at that point in time. If so desired, sender 110 may also receive a copy of the submitted electronic message that was used by EPM system 120 to generate the EPM data structure.

Fig. 2A shows a more detailed embodiment of a system consistent with the present invention. Sender 110 generates an electronic message and sends message data 215, along with an electronic recipient address 220, to an authentication server, called an EPM server 210, over a network 205. Sending the electronic message can be done with an e-mail program, such as, for example, Outlook Express™, or may be done by other methods known to those skilled in the art. Preferably, EPM server 210 is a workstation-class computer, such as, for example, an Intel-based workstation running Windows NT® 4.0. However, other data processing machines known to those skilled in the art may also be used. In this embodiment, EPM server 210 may be a standalone server which accepts data over a network from any external source. Network 205 is preferably a TCP/IP based network which is part of the Internet, but could be, for example, a local area network, Virtual Private Network, a wireless network, and/or any other type of computer network known to those skilled in the art.

EPM server 210 generates an EPM data structure 240 that includes a digest, a temporal-stamp, and a digital signature as described above. After EPM data structure 240 has been generated, EPM server 210 will typically forward the EPM data structure with the electronic message to recipient 130 over network 205. Alternatively, sender 110 may choose to only have EPM data structure 240 forward by EPM server 210 to recipient 130.

Fig. 2B depicts the second embodiment of the invention wherein sender 110 and recipient are the same entity. Sender 110 prepares an electronic message and submits message data 215 and an electronic address through network 205 to EPM server 210 in the same manner as described for Fig. 2. However, in this mode, a sender electronic address 225, rather than recipient electronic address 220 (not shown), is provided with message data 215. EPM server 210 generates an EPM data structure as in the first embodiment, and then submits the EPM back to sender 110. Alternatively, sender 110 may have a copy of message data 215 returned with EPM data structure 240 if desired.

Fig. 3A illustrates a third embodiment consistent with the invention. Sender 110 may consist of a user and a networked device. The networked device will typically be a personal computer. Other examples of a networked device include, but are not limited to, Personal Digital Assistants (PDAs), cell phones, dedicated network terminals, a network server, and other types of electronic devices known to those skilled in the art. It should be noted that the entity which creates the electronic message need not be a human user. Some electronic messages may be generated automatically by computer and submitted for EPMs at predefined times. For example, a business may program a computer to automatically submit electronic bills to customers on a monthly basis through a network and wish to have EPMs generated for these submissions. The embodiment of Fig. 3A is similar to the embodiment of Fig. 2A, except that client entities 310 and 320 are between sender 110 and EPM server 210, and EPM server 210 and recipient 130, respectively. Client entities 310 and 320 provide protection for securing EPM server 210 against unauthorized access and process data to allow EPM server 210 to accept and provide data in standardized formats.

Referring again to Fig. 3A, sender 110 submits a request in the form of an electronic message to a sender client 310 to obtain an EPM. Sender client 310 may be a separate data processing machine, such as, for example, a personal computer or an Intel-based workstation using Windows NT® 4.0 as an operating system.. Alternatively, sender client 310 may be a collection of software modules residing in the networked device of sender 110. In the event sender client 310 is a separate machine, it will receive the request for an EPM over network 205. After sender client 310 accepts the request from sender 110, it processes the request (in a manner described below in greater detail) and transfers the results to EPM server 210 over a secure network 305. If sender client 310 is in close proximity to EPM server 210, secure network 305, could a Local Area Network (LAN) which uses TCP/IP, or another network protocol known to those skilled in the art. If sender client 310 is not in close proximity to EPM server 210, secure network 305 may be Virtual Private Network (VPN) communicating in a secure manner over the Internet. While only one EPM server 210 is shown in Fig. 3A, multiple servers could be used to provide additional reliability. In this embodiment, EPM server operates in a secure environment whereby it has no insecure connection to an external network such as the Internet.

Further referring to Fig. 3A, once EPM server 210 receives the processed request from sender client 310, it generates an EPM data structure and forwards the EPM data structure and recipient electronic address data to a recipient client 320 over secure network 305. Recipient client 320 may be a separate data processing machine located practically at any distance from EPM server 210.

Like sender client 310, it may be, for example, a personal computer or Intel-based workstation using Windows NT® 4.0 as an operating system. Alternatively, recipient client may be a collection of software modules residing at recipient 130. If recipient client 320 is a separate data processing machine, it forwards the EPM data structure to recipient 130 over network 205. Recipient 130 may comprise a user and data processing machine, as described for sender 110 above, or may be a computer, only, automatically processing the received EPMs. Once the EPM data structure is received, recipient 130 may authenticate the EPM data structure and the identity of the signer (the EPM server 210) and, if the corresponding electronic message is available, the electronic message itself. This authentication process may occur at the data processing machine of recipient 130, or it may be performed by a separate data processing machine.

In order to properly authenticate the received EPM data structure, a data processing machine typically requires four elements: a verifier application, the EPM data structure, the electronic message, and an authorized public digital key. EPM server 210 uses a digital signature algorithm to digitally sign EPM data structures. The digital signature is based on public and private digital key pairs. Digital certificates authorize the use of these key pairs used to generate and verify the digital signature. The key authorization process is performed by a Key Signing Authority (KSA) or a Certificate Authority (CA) which issues the digital

LAW OFFICES
FINNEGAN, HENDERSON, FARABOW, GARRETT, & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

certificates. These are trusted, separate third party systems which are not directly coupled to EPM server 210. The KSA is discussed in U.S. serial application no. 60/157,168, filed 9/30/99, and the CA is discussed in U.S. serial application no. 60/189,983, filed 3/17/2000, the entire disclosures of which are incorporated by reference. The authorized public digital key may exist on a physical media in the personal possession of the recipient user 130, or it may be embedded in the verifier software or the EPM data structure itself.

The verifier application performs three verification steps. The first step verifies that the EPM data structure is "official;" that is, it was generated by an authorized entity such as the USPS. It does this by checking the digital certificate associated with the public digital key used to generate the digital signature. When this verification is complete, recipient 130 has proof that the EPM data structure was issued by an official EPM entity.

The second verification step is to verify the identification of the EPM server 210. The digital signature that was used to sign the EPM data structure is verified using the authorized public digital key. When this verification is successful, recipient 130 has proof that the EPM data structure was generated by a particular authorized server (i.e., the identification of EPM server 210 is known) and that alterations to the contents of the EPM data structure, from the time it was generated until the time it reached recipient 130, are detectable. This

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

effectively authenticates the digest and temporal stamp within the EPM data structure.

The third verification step is to authenticate the contents of the electronic message. The verifier application does this by comparing the digest of the message contained in the EPM data structure with a digest generated by the verifier application using the electronic message itself. If the two digests are identical, recipient 130 has proof the contents of the electronic message were unaltered from the time the EPM data structure was generated until the time the EPM data structure and electronic message were received by recipient 130.

Digital signature and electronic message verification functionality can be integrated into platform-independent verifier software which may be downloaded from the Internet. For example, such software may be developed into a Java applet for use in a web browser such as Netscape®, or it could be integrated into an e-mail software application such as Outlook® Express. Alternatively, the verifier application could take the form of an independent software application, such as, for example, a stand alone Windows-based verification utility. The verifier application can make use of standard Application Programming Interfaces (APIs) to provided authentication functionality to software developers in a convenient manner.

Fig. 3B shows a fourth embodiment consistent with the invention. This embodiment is identical to the embodiment of Fig. 3A except here sender client 310 and recipient client 320 are implemented in either the same data processing machine or in the same collection of software modules located, for example, in a networked device at sender 110. In the event they are located in the same data processing machine, sender 110 submits a request in the form of an electronic message to sender/recipient client 310 over network 205. Sender/recipient client 310 accepts and process the request and passes results to EPM server 210 over secure network 305. EPM server 210 generates an EPM data structure and returns the EPM data structure and the electronic address of sender 110 over secure network 305 to sender/recipient client 310. The EPM data structure is then returned to sender 110. Alternatively, EPM server 210 may also return a copy of the electronic message with the EPM data structure. Details of the data transfers between the components in this embodiment are described below in the explanation for Fig. 4B.

Fig. 3C shows a fifth embodiment consistent with the invention. This embodiment is a hybrid of the embodiments of Fig. 3A and 3B. Here, sender 110 submits a request in the form of an electronic message over network 205, to sender/recipient client 310, which in this embodiment is typically a separate data processing machine. Sender/recipient client 310 accepts and process the request and passes results to EPM server 210 over secure network 305. EPM server 210 generates an EPM data structure and returns the EPM data structure and the electronic address of sender 110 over secure network 305 to sender/recipient client 310. Sender/recipient client 310 then forwards the message and EPM data structure to recipient 130 over network 205. Details of the data transfers between the components in this embodiment are described below in the explanation for Fig. 4C.

Fig. 4A depicts the data flow for the embodiment of Figure 3A. Sender 110 generates an electronic message containing message data 215 which may be in any type of format. For example, message data 215 could be a clear-text ASCII file or encrypted ASCII file, a raw binary file, or a text-encoded binary file using base64 or other binary-to-text encoding method known to those skilled in the art.

Message data 215 is bundled with a recipient electronic address 220. If sender client 310 resides in a separate data processing machine, the bundle is sent over network 205 to sender client 310. Otherwise, if sender client 310 exists as a collection of software modules residing on the networked device of sender 110, message data 215 and recipient electronic address 220 are sent over secure network 305. Message data 215 and recipient electronic address 220 will typically be sent using an e-mail program, such as, for example, Outlook

Express, running on a personal computer at sender 110. However, other types of file transfer programs using different transport protocols, such as WinFTP, may also be used.

Sender client 310 then produces a hash value 420 from message data 215 using a one-way hash function. As known to those skilled in the art, a one-way hash function typically generates a hash value from input data which is substantially smaller than the input data itself. The hash value is generated by an algorithm such that the probability of two different data streams producing the same hash value is extremely small; in fact so small that the hash value is considered unique to the input data. The one-way hash function cannot be reversed; the input data itself cannot be recovered from its corresponding hash value. Hash value 420 is thus a unique number associated on a one-to-one basis with message data 215.

Sender client 310 packages hash value 420 with recipient electronic address 220 and transfers them to EPM server 210 over secure network 305. Optionally, message data 215 may also be passed along in this transfer. A time-stamp and/or a date-stamp are generated by EPM server 210 and bundled with hash value 420. EPM server 210 then generates a digital signature using a Digital Signature Standard algorithm which is known to those skilled in the art. It then applies the digital signature to the bundled data to form an EPM data structure 240. Additional branding data, discussed in more detail below, may also be included in EPM data structure 240.

Recipient client 320 receives EPM data structure 240 and recipient electronic address 220 over secure network 305 from EPM server 210. Recipient client 320 uses recipient electronic address 220 to send EPM data structure 240 to recipient 130. If recipient client 320 is a separate data processing machine, it may use network 205 for the transfer. If recipient client 320 is a collection of software modules, for example contained in recipient 130, it typically uses secure network 305 for the transfer. Optionally, recipient 130 may also receive message data 215 itself, along with EPM data structure 240 through the recipient client 320, if sender 110 decides to route message data 215 through EPM server 210. Note that this alternative data flow is shown in the dashed boxes in Fig. 4A.

Fig. 4B illustrates the data flow for the embodiment of Fig. 3B. The data flow from sender 110 to EPM server 210 is identical to that described for Fig. 4A. In this embodiment, EPM server 210 returns EPM data structure 240 along with recipient electronic address 220, which in this case is the electronic address of sender 110, to sender/recipient client 310 over secure network 305. Using recipient electronic address 220, sender/recipient client 310 forwards the EPM data structure 240 to sender 110 over network 205.

Similarly to the data flow shown in Fig. 4A, a copy of message data 215 itself, along with EPM data structure 240, may be returned to sender 110 if desired by the sender. In this instance, message data 215 may be routed via EPM sever 210. Another option is to match EPM data structure 240 with message data 215 on sender/recipient client 310, thus obviating the need to forward message data 215 to EPM server 210. Note that these alternative data flow options are shown by the dashed boxes in Fig. 4B.

Fig. 4C illustrates the data flow for the embodiment of Fig. 3C. The data flow from sender 110 to sender/recipient client 310 is identical to that described for Fig. 4A. In this embodiment, sender/recipient client 310 forwards hash value 420 and recipient address 220 to EPM server 210. EPM server 210 generates an EPM data structure 240 and returns this data structure, along with recipient electronic address 220, to sender/recipient client 310 over secure network 305. Using recipient electronic address 220, sender/recipient client 310 forwards EPM data structure 240 and message data 215 to recipient 130. EPM data structure 240 and message data 215 will typically be sent over network 205.

Fig. 5A shows the software modules responsible for inter-client interaction for sender client 310, EPM server 210, and recipient client 320 for the embodiment of Fig. 3A. These modules comprise collections of software routines for execution on a data processing machine. The sender client consists of two major modules, front-end module 510 and client proxy module 520. Front-end module 510 receives requests for the generation of an EPM over network 205. Client proxy module 520 includes a network client module 530 and presents a class interface, preferably written in, but not limited to, C++, which is instantiated by front-end module 510 to handle the submission of a request for an EPM for a given message data content. Client proxy module 520 uses networking services contained in network client module 530 to send EPM transaction requests and associated data to EPM server 210. Preferable networking services use the TCP/IP standards, however, the invention is not limited to any networking protocol. Network client module 530 sends the transaction requests over secure network 305 to EPM server 210 where they are queued for subsequent processing.

EPM server 210 generates an EPM data structure 240 in response to a request and places it in an outgoing queue with any associated data for transmission over secure network 305 to recipient client 320. Recipient client 320 also includes a network client module 540 which receives EPM data structure 240 and recipient electronic address 220 and routes it to a front-end module 560 via a client proxy module 550. Front end module 560 sends the EPM data structure 240 and associated data to recipient 130 via network 205.

Fig. 5B depicts the software modules responsible for inter-client interaction for the embodiment of Fig. 3B. In Fig. 5B, front-end module 510, client proxy module 520, and network client module 530 perform the identical functions of modules of the same reference numbers shown in Fig. 5A and described above. In addition, front-end module 510, client proxy module 520, and network client module 530 also perform the functions of front-end module 560, client proxy module 550, and network client module 540 shown in Fig. 5A and described above.

Fig. 6 illustrates hardware and software components of EPM server 210. EPM server 210 provides a trusted and reliable service for the authentication of electronic messages. EPM server 210 thus protects electronic messages with the legal mechanisms which currently protect official entities such as the USPS. EPM server 210 is thus preferably designed, constructed, and operated as a secure computing system within an environment completely controlled by the official entity, or a trusted delegate of the official entity.

EPM Server 210 preferably comprises a fully multi-threaded server that accepts transactions from an external source, generates an EPM data structure 240, and forwards EPM data structure 240 to the entity that made the request. EPM server 210, upon receiving a sender client 310 connection, spawns a new thread performs all communication functions with sender client 310. When sender client 310 transmits a request, the new thread collects the entire request from sender client 310 then place this request on an input queue. When a thread associated with generating the actual EPM locates the request in the input queue, it flags the request as "In Progress" and then proceeds to produce an EPM data structure 240. Once the EPM generation is complete, EPM data structure 240 is placed on the outbound queue for pickup by recipient client 320, as shown in the embodiment of Fig. 3A. Alternatively, as shown in the embodiment of Fig. 3B and Fig. 3C, EPM data structure 240 may be picked up by sender/recipient client 310.

Network server module 550 includes a TCP/IP module 610 and a transaction processor module 615. Note the invention is not limited to using TCP/IP, however this standard is the preferred network protocol. Transaction processor module 615 uses the services of TCP/IP module 610 to process EPM request messages from sender client 310. Transaction processor module 615 receives each incoming EPM request and forwards it to an EPM module 620 for postmark processing. When the EPM module 620 returns the generated EPM data structure 240, transaction processor module 615 sends EPM data structure 240 it to either sender 110 or recipient 130, based upon the incoming request.

EPM module 620 uses the services of a time module 625 to obtain highly accurate time-stamps for EPMs. In addition to time and date information, EPM module 620 may gather other data items, including branding data, from a system registry 660. Branding data, which contains information regarding the corporate or organizational entity which operates EPM server 210, can take the form of text or image data. This data may represent names, slogans, logos, or any other type of identifying information, and may be included with hash value 420 and the temporal stamp.

Services from a cryptographic interface module 640 are used to generate a digital signature based on the hash value and temporal stamp to create EPM data structure 240. When the EPM operation is deemed complete, EPM module 620 uses a log module 665 to create an entry into a log file 667 which contains copies of each EPM data structure 240 generated. Log file 667 can be used for audit and billing purposes, and provides legal proof that a given EPM data structure 240 was generated. Given the importance of the log file in the audit process, the entire log file itself is used as the input data for generation of a log file EPM data structure 240 to ensure its integrity. The log file EPM data structure may be generated automatically, based on a number of criteria including the size of log file 667 or a fixed interval of time between log file EPMs. This operation may also be done manually at the command of the operator of EPM server 210.

One or more hardware clocks 635 are used to obtain and maintain accurate and trusted time information. The time values are typically generated and stored using Universal Time Coordinated (UTC), which is the same as Greenwich Mean Time. By way of example only, a TrueTime model PCI-SG Synchronized Clock Generator with GPS (Global Positioning System), commercially available from TrueTime, Inc. of Santa Rosa, CA, may be used which has UTC accuracy of approximately one microsecond. A time manager Graphical User Interface (GUI) module 630 allows an operator to set and reset time-stamp information, synchronize time module 625 with hardware clock(s) 635, and visually check the correctness of time from hardware clock(s) 635.

Cryptographic interface module 640 uses one or more hardware cryptographic devices 645 to perform digital signature generation and verification, key generation, and hashing functions. Cryptographic hardware device 645 is able to support multiple encryption algorithms. By was of example only, an Attalla Websafe/PCI card commercially available from Compaq Corporation of Houston, Texas, may be used for cryptographic device 645. Furthermore, the Digital Signature Algorithm (DSA) with the option of Elliptic Curve DSA may be used for the digital signature algorithms. Moreover, EPM server 210 may generate Digital Signature Standard (DSS) keys and use the Secure Hash Standard FIPS 180-1, and the DSS FIPS 186. All of these examples are for exemplary purposes only, and are not meant to limit the present invention.

Cryptographic interface module 640 is controlled by a key manager GUI 650 in order to allow the generation of new digital key pairs for use by EPM server 210 and for the export of unauthorized public digital keys. Key manager GUI allows an EPM security officer to choose a location to store the unauthorized public digital key for a new key pair. Once a new pair is generated, the unauthorized public digital key of the pair is transmitted to a KSA or CA in order to transform it into an authorized public digital key. The private key of the key pair is stored within EPM server 210 and typically is not exported. The unauthorized public digital key is taken to a KSA or CA through secure mechanisms such as actual physical transport by authorized personnel or over a network secured by using encryption techniques. The resulting authorized digital key may be stored within the EPM server for inclusion into digital signatures, may be embedded into the verifier application residing on the data processing machine responsible digital signature authentication, or may be placed on a physical medium and kept by the user at recipient 130.

EPM server 210 supports a configuration manager GUI 665 that allows EPM server 210 system parameters to be set at the time of initialization and setup. This GUI may also be used to thereafter to update the configuration parameters of an operational EPM server. These system parameters are changed by accessing values stored in system registry 660.

Fig. 7 is a detailed block diagram showing the components corresponding to sender client 310, EPM server 210, and recipient client 320. Sender client 310 contains a solid-state memory 710 which holds instructions which are transferred over a bus 715 for execution by a CPU 725. Memory 710 contains an operating system 711, such as, for example, Windows™ NT 4.0 Workstation or Unix clients. Also included in memory 710 are front-end module 510, client-proxy module 520, and network client module 530. Instructions of these modules are also contained in mass storage device 720, and are loaded into memory 710 in whole or in part during initialization of sender client 310. Also connected to bus 715 are user input device interface 730 and user output device interface 735. Sender client 310 communicates over network 205 and secure network 305 through network device interface 740.

Recipient client 320 may have the same hardware configuration as sender client 310. Memory 791 will contain operating system 792, front-end module 560, client proxy module 550, and network client module 540. Each of the modules includes the same function as its counterpart found in memory 710 of sender client 310.

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

EPM server 210 contains a solid-state memory 747 which holds instructions which are transferred over a bus 754 for execution by a CPU 755. Memory 747 contains an operating system 748, such as, for example, Windows NT® 4.0 Server or Unix. Also included in memory are network server module 550, log module 655, EPM module 620, cryptographic interface module 640, configuration manager GUI 665, time manager GUI 630, and key manager GUI 650. These instructions are also contained in mass storage device 750, and are loaded into memory 710 in whole or in part during initialization of EPM server 210. Also contained in mass storage 750 is system registry 660 and log file 667. Connected to bus 715 are user input device interface 760 and user output device interface 765. Cryptographic device 645 and hardware clocks 635 are also connected to bus 754 to allow communication with appropriate software modules residing in memory 747. EPM server 210 communicates over secure network 305 through network device interface 780.

Fig. 8A illustrates the processing steps which occur to produce EPM data structure 240. A one-way hash function is performed using message data 215 to produce a digest, or hash value 420. Preferably, the one-way hash function is performed by client sender 310, but may be generated by the EPM server 210. Hash value 420 is bundled with the time and date stamp 810, obtained from time module 625. Optionally, branding information can also be included in this bundle

LAW OFFICES
FINNEGAN, HENDERSON,
FARABOW, GARRETT,
& DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005
202-408-4000

which may present information regarding the organization offering the EPM service as described above. Furthermore, a value uniquely identifying each EPM data structure can also be included. This value can be used to facilitate account purposes. The hash value, time and date stamp, branding data, and identifier value may then "sealed" or secured through a digital signature.

As known to those skilled in the art, digital signature 820 may be produced by first performing a secure hash algorithm by using, for example, the Secure Hash Standard FIPS 180-1 on the data to be signed to produce a secure hash value. The secure hash value is then processed using a digital signature algorithm (DSA) and a unique private key to produce two data values. These data values comprise digital signature 820, which is appended to the hash value, time and data stamp, and branding data to form EPM data structure 240.

In order to validate the digital signature, a public digital key, which has a unique pairing with the private key, must be used. Methods known to those skilled in the art, such as, for example, the Digital Signature Standard, may be used to produce digital signature 820.

Fig. 8B shows an alternate method of forming EPM data structure 240. The process is similar to that described above for Fig. 8A. However, in the method of Fig. 8A, an embedded authorized public digital key 830 is included with the digital signature 820. This method has the advantage of not requiring a key at recipient 130 in order to authenticate digital signature 820. However, to maintain the security of EPM system 120, embedded digital key 830 should only be used on a one-time basis, i.e., embedded digital key 830 should only be able to authenticate the digital signature for the single electronic message EPM data structure 240 is associated with. Additional messages sent to the same recipient 130 would include a unique embedded authorized digital key for each EPM data structure 240 sent.

The foregoing description is presented for purposes of illustration and explanation. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and modifications of variations are possible in light of the above teachings or may be acquired from practice of the invention. The principles of the invention and its practical application enable one skilled in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated.